AMENDMENTS TO THE CLAIMS

The following listing of claims will replace all prior versions and listings of claims in the application:

1. (Previously Canceled)

2. (Previously Presented): A processor comprising:

an execution unit to execute instructions;

a replay system coupled to the execution unit to replay instructions which have not executed properly, the replay system comprising:

a checker to determine whether each instruction has executed properly;

a replay queue coupled to the checker to temporarily store one or more instructions for replay;

a replay loop to route an instruction which executed improperly to an execution unit for replay; and

a replay queue loading controller to determine whether to load an improperly executed instruction to the replay loop or into the replay queue.

3. (Original): The processor of claim 2 and further comprising:

a scheduler to output instructions; and

a multiplexer or selection mechanism having a first input coupled to the scheduler, a second input coupled to the replay loop and a third input coupled to an output of the replay queue.

4. (Canceled).

5. (Canceled).

6. (Previously Presented) A processor comprising:

an execution unit to execute instructions;

a replay system coupled to the execution unit to replay instructions which have not executed properly, the replay system comprising:

a checker to determine whether each instruction has executed properly; and

a replay queue coupled to the checker to temporarily store one or more instructions for replay, wherein the replay queue comprises a replay queue coupled to the checker to temporarily store an instruction in which source data must be retrieved from an external memory device, the instruction being unloaded from the replay queue when the source data for the instruction returns from the external memory device.

7. (Previously Presented): The processor of claim 2 wherein said execution unit is a memory load unit, the processor further comprising:

a first level cache system coupled to the memory load unit;

a second level cache system coupled to the first level cache system; and

wherein the memory load unit performs a data request to external memory if there is a miss on both the first level and second level cache systems.

8. (Original): The processor of claim 7 wherein a load instruction will be loaded into the replay queue when there is a miss on both the first level and second level cache systems, and the load instruction is unloaded from the replay queue for re-execution when the data for the instruction returns from the external memory.

9. (Original): A processor comprising:

a multiplexer having an output;

a scheduler coupled to a first input of the multiplexer;

an execution unit coupled to an output of the multiplexer;

a checker coupled to the output of the multiplexer to determine whether an instruction has executed properly;

a replay queue to temporarily store instructions, an output of the replay queue coupled to a second input of the multiplexer; and

a controller coupled to the checker to determine when to load an instruction into the replay queue and to determine when to unload the replay queue.

10. (Original): The processor of claim 9 and further comprising a staging section coupled between the checker and a third input to the multiplexer to provide a replay loop, the controller controlling the multiplexer to select either the output of the scheduler, the replay loop or the output of the replay queue.

11. (Original): The processor of claim 9 wherein the controller loads an instruction into the replay queue when the instruction is not ready to execute properly, and unloads the instruction from the replay queue when the instruction is ready to execute properly.

12. (Original): The processor of claim 9 wherein the controller determines when to unload the replay queue based on a data return signal.

13—20. (Previously Canceled)

21. (Canceled).

22. (Canceled).

23. (Previously presented): The processor of claim 6 wherein a controller determines whether to unload the replay queue based on a data return signal.

24. (New) The processor of claim 2, further including a scoreboard coupled to the checker, the scoreboard to track whether source data of each instruction is correct prior to a time of execution.

25. (New) The processor of claim 24, wherein the checker is to determine that an instruction has executed improperly if the source data is incorrect at the time of execution.

26. (New) The processor of claim 24, wherein the checker is to determine that an instruction has executed properly if the source data is correct at the time of execution.

27. (New) The processor of claim 3, wherein the scheduler is to create an open slot in an instruction stream output from the multiplexer in response to a stop scheduler signal.

28. (New) The processor of claim 27, wherein the checker is to generate the stop scheduler signal.

30. (New) The processor of claim 6, further including a scoreboard coupled to the checker, the scoreboard to track whether source data of each instruction is correct prior to a time of execution.

31. (New) The processor of claim 30, wherein the checker is to determine that an instruction has executed improperly if the source data is incorrect at the time of execution.

32. (New) The processor of claim 30, wherein the checker is to determine that an instruction has executed properly if the source data is correct at the time of execution.

33. (New) The processor of claim 6, wherein the replay queue is to further store instructions depending upon the instruction in which source data must be retrieved from the external memory device.

34. (New) The processor of claim 33, wherein the instructions depending upon the instruction in which source data must be retrieved from the external memory device are to be defined as improperly executing instructions that are programmatically younger than the instruction in which source data must be retrieved from the external memory device.

35. (New) The processor of claim 6, wherein the instruction in which source data must be retrieved from the external memory device is a long latency instruction.

36. (New) A system comprising:

a memory; and

a processor coupled to the memory, the processor including an execution unit to execute instructions and a replay system coupled to the execution unit to replay instructions which have not executed properly, the replay system having a checker to determine whether each instruction has executed properly and a replay queue coupled to the checker to temporarily store one or more long latency instructions until the long latency instruction is ready for execution.

37. The system of claim 36, wherein the replay system further includes:

a replay loop to route an instruction which executed improperly to the execution unit for replay; and

a replay queue loading controller to determine whether to load an improperly executed instruction to the replay loop or into the replay queue.

38. The system of claim 36, wherein the long latency instruction is to be unloaded from the replay queue if the instruction is ready to execute properly.

39. The system of claim 38, wherein the replay system is to generate a replay queue select signal in response to the instruction being ready to execute properly.

40. The system of claim 36, wherein source data is to be retrieved from the disk memory, the instruction to be unloaded from the replay queue if the source data for the instruction returns from the memory.